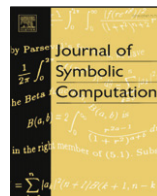




Contents lists available at ScienceDirect

Journal of Symbolic Computation

journal homepage: www.elsevier.com/locate/jsc

Algorithm computing the local b function by an approximate division algorithm in $\hat{\mathcal{D}}$

Hiromasa Nakayama

Graduate School of Science and Technology, Kobe University, Kobe 657-8501, Japan

ARTICLE INFO

Article history:

Received 30 November 2006

Accepted 14 June 2007

Available online 26 September 2008

Keywords:

Division in the ring of analytic differential operators

Mora division algorithm

 b -functionLocal b -function

ABSTRACT

We give an approximate division algorithm in the ring of analytic differential operator $\hat{\mathcal{D}}$. By using this approximate division algorithm and the Mora division algorithm for differential operator, we can compute the local b function.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Let K be a field of characteristic 0. We denote the differential operator $\frac{\partial}{\partial x_i}$ by ∂_i . The ring of differential operators with polynomial coefficient

$$D = \left\{ \sum_{\beta \in \mathbb{N}^n} a_{k,\beta}(x) \partial^\beta \mid a_{k,\beta}(x) \in K[x] \right\}$$

is denoted by D , and that with formal power series coefficient

$$\hat{\mathcal{D}} = \left\{ \sum_{\beta \in \mathbb{N}^n} a_{k,\beta}(x) \partial^\beta \mid a_{k,\beta}(x) \in K[[x]] \right\}$$

is denoted by $\hat{\mathcal{D}}$. Here $x = (x_1, \dots, x_n)$ and $\partial = (\partial_1, \dots, \partial_n)$.

E-mail address: nakayama@math.kobe-u.ac.jp.

Division theorems are fundamental in several construction for D -modules. Castro gave a constructive division theorem in $\widehat{\mathcal{D}}$, which gives unique quotients and remainder (Castro, 1984). However, the division procedure needs infinite reductions. On the other hand, Granger, Oaku and Takayama gave a division algorithm for algebraic data in $\widehat{\mathcal{D}}$ (Granger and Oaku, 2004; Granger et al., 2005), which is an analogous algorithm with the Mora division algorithm in the polynomial ring (Greuel and Pfister, 1996). We call this division the Mora division algorithm in D . The division algorithm stops in finite steps, but the remainder is not completely reduced. By using the division algorithm, we can get a Gröbner basis and solve the ideal membership problem for algebraic data in $\widehat{\mathcal{D}}$.

The local b function of a polynomial $f \in K[x]$ is the minimum degree monic polynomial $b(s) \in K[s]$ which satisfies

$$\exists P \in \widehat{\mathcal{D}}[s] \quad \text{such that } Pf^{s+1} = b(s)f^s.$$

Oaku gave algorithms computing the local b function of a given polynomial by using a Gröbner basis method in D (Oaku, 1997b,a; Oaku and Takayama, 2001). In this paper, we propose a new algorithm to compute the local b function. We use the Mora division algorithm in D and an approximate division algorithm in $\widehat{\mathcal{D}}$, which gives an approximation of the remainder by Castro's division in $\widehat{\mathcal{D}}$.

2. Division theorem and approximate division algorithm in the ring of power series $K[[x]]$

There are several kinds of division theorem in $K[[x]]$. Among them, we need a division theorem which gives unique quotients and unique remainder for our approximate division algorithm in $\widehat{\mathcal{D}}$. We start with explaining this kind of division theorem. As to details about the division theorem including a history, we refer Castro and Granger (2004).

We define a monomial order $<_r$ on monomials in $K[[x]]$ by the following $1 \times n$ matrix and a term order $<$ as the tie-breaker.

$$\begin{array}{ccc} x_1 & \cdots & x_n \\ -1 & \cdots & -1 \\ \hline & \text{term order } < & \end{array}$$

In other words, we define the order $<_r$ as

$$x^\alpha <_r x^\beta \Leftrightarrow \begin{cases} -\alpha_1 - \cdots - \alpha_n < -\beta_1 - \cdots - \beta_n \text{ or} \\ (-\alpha_1 - \cdots - \alpha_n = -\beta_1 - \cdots - \beta_n \text{ and } x^\alpha < x^\beta). \end{cases}$$

Since the order is not a well order, ordinary division algorithm does not generally stop in finite steps.

Let $\alpha(1), \dots, \alpha(s) \in (\mathbb{Z}_{\geq 0})^n$. For $(\alpha(1), \dots, \alpha(s))$, we define a partition $\Delta^1, \dots, \Delta^s, \overline{\Delta}$ of $(\mathbb{Z}_{\geq 0})^n$ as

$$\begin{aligned} \Delta^1 &= \alpha(1) + (\mathbb{Z}_{\geq 0})^n, \\ \Delta^2 &= (\alpha(2) + (\mathbb{Z}_{\geq 0})^n) \setminus \Delta^1, \dots \\ \Delta^s &= (\alpha(s) + (\mathbb{Z}_{\geq 0})^n) \setminus (\Delta^1 \cup \cdots \cup \Delta^{s-1}), \\ \overline{\Delta} &= (\mathbb{Z}_{\geq 0})^n \setminus (\Delta^1 \cup \cdots \cup \Delta^s). \end{aligned}$$

Let $f \in K[[x]]$. We define the following notations.

$\text{LM}_{<_r}(f)$: leading monomial of f with respect to $<_r$

$\text{LT}_{<_r}(f)$: leading term of f with respect to $<_r$

$\text{LE}_{<_r}(f)$: leading exponent of f with respect to $<_r$

$\text{Exps}(f)$: set of exponents appearing in f

$\text{Rest}_{<_r}(f) : f - \text{LT}_{<_r}(f).$

For example, let $f = 3x_1 + x_1x_2$, then $\text{LM}_{<_r}(f) = x_1$, $\text{LT}_{<_r}(f) = 3x_1$, $\text{LE}_{<_r}(f) = (1, 0)$, $\text{Exps}(f) = \{(1, 0), (1, 1)\}$, and $\text{Rest}_{<_r}(f) = x_1x_2$.

Lemma 1 (Division by Terms in $K[[x]]$). Let $f \in K[[x]]$, $g_1, \dots, g_s \in K[[x]]$ and $\Delta^1, \dots, \Delta^s, \overline{\Delta}$ be the partition of $(\mathbb{Z}_{\geq 0})^n$ with respect to $(\text{LE}_{<_r}(g_1), \dots, \text{LE}_{<_r}(g_s))$. Then, there exist quotients $q_1, \dots, q_s \in K[[x]]$ and remainder $r \in K[[x]]$ which satisfies the following conditions:

$$f = q_1 \text{LT}_{<_r}(g_1) + \dots + q_s \text{LT}_{<_r}(g_s) + r,$$

$$\text{Exps}(g_i) + \text{Exps}(q_i) \subset \Delta^i,$$

$$\text{Exps}(r) \subset \overline{\Delta}.$$

Namely, this is a division of f by leading terms $\text{LT}_{<_r}(g_1), \dots, \text{LT}_{<_r}(g_s)$. Especially, (q_1, \dots, q_s, r) is uniquely determined by $(f, \text{LT}_{<_r}(g_1), \dots, \text{LT}_{<_r}(g_s), <_r)$. We call this computation **term-div** (f, g_1, \dots, g_s) .

Theorem 2 (Division Theorem in $K[[x]]$). Let $f \in K[[x]]$, $g_1, \dots, g_s \in K[[x]]$, and $\Delta^1, \dots, \Delta^s, \overline{\Delta}$ be the partition of $(\mathbb{Z}_{\geq 0})^n$ with respect to $(\text{LE}_{<_r}(g_1), \dots, \text{LE}_{<_r}(g_s))$. Then, there exist quotients $q_1, \dots, q_s \in K[[x]]$ and remainder $r \in K[[x]]$ which satisfies the following conditions.

$$f = q_1 g_1 + \dots + q_s g_s + r$$

$$\text{LE}_{<_r}(g_i) + \text{Exps}(q_i) \subset \Delta^i$$

$$\text{Exps}(r) \subset \overline{\Delta}.$$

Especially, (q_1, \dots, q_s, r) is uniquely determined by $(f, g_1, \dots, g_s, <_r)$.

We show a procedure to obtain q_1, \dots, q_s, r .

$K[[x]]$ -division(f, g_1, \dots, g_s)

Input : $f \in K[[x]]$, $g_1, \dots, g_s \in K[[x]]$
Output : $q_1, \dots, q_s, r \in K[[x]]$ which satisfies

$$f = q_1 g_1 + \dots + q_s g_s + r$$

$$\text{LE}_{<_r}(g_i) + \text{Exps}(q_i) \subset \Delta^i$$

$$\text{Exps}(r) \subset \overline{\Delta}$$

```

1 :   $F \leftarrow f, q_i \leftarrow 0$  ( $1 \leq i \leq s$ ),  $r \leftarrow 0$ 
2 :  while ( $F \neq 0$ ) {
3 :     $[q', r'] \leftarrow \text{term-div}(F, g_1, \dots, g_s)$ 
4 :     $q_i \leftarrow q_i + q'_i$  ( $1 \leq i \leq s$ )
5 :     $r \leftarrow r + r'$ 
6 :     $F \leftarrow -\sum_{i=1}^s q'_i \text{Rest}_{<_r}(g_i)$ 
7 :  }
8 :  return  $[q, r]$ 

```

This procedure generally does not stop in finite steps. Therefore, this procedure is not an algorithm in a strict sense. If input f and g_i are polynomials, we can give an algorithm which returns approximate quotients and approximate remainder which are correct up to given total degree N . Approximate algorithm of this kind has been discussed by several authors, for example, see Kobayashi et al. (1986). After the manner of the approximate algorithm, we will give an algorithm which gives approximations of quotients and remainder by $K[[x]]$ -division.

Algorithm 1 ($K[[x]]$ -Approximate-Division).

 $K[[x]]$ -approximate-division(f, g_1, \dots, g_s, N)

Input : $f \in K[x], g_1, \dots, g_s \in K[x], N \in \mathbb{Z}_{>0}$ Output : $\bar{q}_1, \dots, \bar{q}_s, \bar{r}, \bar{F} \in K[x]$ which satisfies the following conditions.

$$f = \bar{q}_1 g_1 + \dots + \bar{q}_s g_s + \bar{r} + \bar{F}$$

$$\bar{F} \neq 0 \Rightarrow |\text{LE}_{<_r}(\bar{F})| \geq N$$

Terms of \bar{q}_i whose total degree is smaller than $N - |\text{LE}_{<_r}(g_i)|$ agree with those of q_i .Terms of \bar{r} whose total degree is smaller than N agree with those of r .Here $|\alpha|$ is $\sum_{i=1}^n \alpha_i$, where $\alpha = (\alpha_1, \dots, \alpha_n)$,and q_i, r are quotients and remainder of $K[[x]]$ -division(f, g_1, \dots, g_s).1 : $\bar{F} \leftarrow f, \bar{q}_i \leftarrow 0$ ($1 \leq i \leq s$), $\bar{r} \leftarrow 0$ 2 : while ($\bar{F} \neq 0$ and $|\text{LE}_{<_r}(\bar{F})| < N$) {3 : $[q', r'] \leftarrow \text{term-div}(\bar{F}, g_1, \dots, g_s)$ 4 : $\bar{q}_i \leftarrow \bar{q}_i + q'_i$ ($1 \leq i \leq s$)5 : $\bar{r} \leftarrow \bar{r} + r'$ 6 : $\bar{F} \leftarrow -\sum_{i=1}^s q'_i \text{Rest}_{<_r}(g_i)$

7 : }

8 : return $[\bar{q}, \bar{r}, \bar{F}]$ **Example 3** (Example of $K[[x]]$ -Approximate-Division). The example of the case of $f = x, g_1 = 1 + x, N = 5$ (execute $K[[x]]$ -approximate-division($x, 1 + x, 5$))

| term-div | q' | r' | \bar{q} | \bar{r} | \bar{F} |
|---------------------------|--------|------|-----------------------------|-----------|-------------------|
| | | | 0 | 0 | x |
| $x = x \cdot 1 + 0$ | x | 0 | x | 0 | $-x \cdot x$ |
| $-x^2 = -x^2 \cdot 1 + 0$ | $-x^2$ | 0 | $x - x^2$ | 0 | $-(-x^2) \cdot x$ |
| $x^3 = x^3 \cdot 1 + 0$ | x^3 | 0 | $x - x^2 + x^3$ | 0 | $-x^3 \cdot x$ |
| $-x^4 = -x^4 \cdot 1 + 0$ | $-x^4$ | 0 | $x - x^2 + x^3 - x^4$ | 0 | $-(-x^4) \cdot x$ |
| $x^5 = x^5 \cdot 1 + 0$ | x^5 | 0 | $x - x^2 + x^3 - x^4 + x^5$ | 0 | $-x^5 \cdot x$ |

The output of the approximate division algorithm is

$$x = (x - x^2 + x^3 - x^4 + x^5) \cdot (1 + x) - x^6.$$

In this case, the quotients and the remainder by $K[[x]]$ -division are

$$\begin{aligned} x &= \frac{x}{1+x} \cdot (1+x) + 0 \\ &= (x - x^2 + x^3 - x^4 + x^5 - \dots) \cdot (1+x) + 0. \end{aligned}$$

3. Division theorem and approximate division algorithm in $\widehat{\mathcal{D}}[y]$

Castro gave a constructive division procedure in $\widehat{\mathcal{D}}$ (Castro, 1984). His division procedure returns unique quotients and remainder. In this section, we give an algorithm which gives approximations of the quotients and the remainder by his division. The approximate quotients and remainder are correct up to a given total degree.

Let y be a parameter, and ξ be a commutative variable which stands for ∂ . We define a monomial order $<$ on $\widehat{\mathcal{D}}[y]$ by using the following $2 \times (2n+1)$ matrix and the tie-breaker $<_1$.

$$\begin{array}{ccccccccc} x_1 & \cdots & x_n & y & \xi_1 & \cdots & \xi_n \\ \hline 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ -1 & \cdots & -1 & 0 & 0 & \cdots & 0 \\ (& & \text{term order} & <_1 &) \end{array}$$

And we define an other monomial order $<_r$ on $\widehat{\mathcal{D}}[y]$ by using the following $1 \times (2n + 1)$ matrix and tie-breaker $<_1$.

$$\begin{array}{ccccccccc} x_1 & \cdots & x_n & y & \xi_1 & \cdots & \xi_n \\ \hline -1 & \cdots & -1 & -1 & -1 & \cdots & -1 \\ \text{(term order } <_1 \text{)} \end{array}$$

We define a weight vector e for monomials $\{x^\alpha \xi^\beta y^\gamma\}$ as

$$\begin{pmatrix} x_1 & \cdots & x_n & y & \xi_1 & \cdots & \xi_n \\ 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \end{pmatrix}$$

If $|\beta| + |\gamma| = |\beta'| + |\gamma'|$ (in other words, monomials have the same order with respect to the weight vector e), then it holds that

$$x^\alpha \xi^\beta y^\gamma < x^{\alpha'} \xi^{\beta'} y^{\gamma'} \Leftrightarrow x^\alpha \xi^\beta y^\gamma <_r x^{\alpha'} \xi^{\beta'} y^{\gamma'}.$$

Therefore, we get

$$\text{LM}_{<}(P) = \text{LM}_{<_r}(\text{in}_e(P)).$$

Here $\text{in}_e(P)$ is the initial part of P with respect to the weight vector e . For example, when $P = x_1 \partial_1^2 + x_1^2 \partial_1^2 + x_1 \partial_1 + 1$, we have $\text{in}_e(P) = x_1 \xi_1^2 + x_1^2 \xi_1^2$. And $\text{LM}_{<_r}(\text{in}_e(P)) = x_1 \xi_1^2$, $\text{LM}_{<}(P) = x_1 \xi_1^2$.

We review Castro's division procedure in $\widehat{\mathcal{D}}[y]$. Of course, since the monomial order $<$ is not a well order, the procedure needs infinite reductions.

Theorem 4 ((Castro, 1984) Division Theorem in $\widehat{\mathcal{D}}[y]$). Let $P, P_1, \dots, P_s \in \widehat{\mathcal{D}}[y]$, and $\Delta^1, \dots, \Delta^s, \overline{\Delta}$ be the partition of $(\mathbb{Z}_{\geq 0})^{2n+1}$ with respect to $(\text{LE}_{<}(P_1), \dots, \text{LE}_{<}(P_s))$. There exist $Q_1, \dots, Q_s, R \in \widehat{\mathcal{D}}[y]$ which satisfies the following conditions.

$$P = Q_1 P_1 + \cdots + Q_s P_s + R$$

$$\text{LE}_{<}(P_i) + \text{Exps}(Q_i) \subset \Delta^i$$

$$\text{Exps}(R) \subset \overline{\Delta}.$$

Epecially, (Q_1, \dots, Q_s, R) are uniquely determined by $(P, P_1, \dots, P_s, <)$.

We show procedure for the division theorem.

$$\begin{array}{l} \widehat{\mathcal{D}}\text{-division}(P, P_1, \dots, P_s) \\ \hline \text{Input : } P, P_1, \dots, P_s \in \widehat{\mathcal{D}}[y] \\ \text{Output : } Q_1, \dots, Q_s, R \in \widehat{\mathcal{D}}[y] \text{ which satisfies} \\ \quad P = Q_1 P_1 + \cdots + Q_s P_s + R \\ \quad \text{LE}_{<}(P_i) + \text{Exps}(Q_i) \subset \Delta^i \\ \quad \text{Exps}(R) \subset \overline{\Delta} \\ \hline 1 : \quad Q_i \leftarrow 0 \quad (1 \leq i \leq s), R \leftarrow P \\ 2 : \quad m_0 \leftarrow \text{ord}_e(R) \\ 3 : \quad \text{for } (k \leftarrow m_0; k \geq 0; k \leftarrow k - 1) \{ \\ 4 : \quad \quad r \leftarrow (\text{total symbol of the part of } R \text{ whose } e\text{-order is } k) \\ 5 : \quad \quad [q'_i, r'] \leftarrow K[[x]]\text{-division}(r, \text{in}_e(P_1), \dots, \text{in}_e(P_s), <_r) \\ 6 : \quad \quad Q'_i \leftarrow (\text{replace } \xi \text{ with } \partial \text{ in } q'_i) \quad (1 \leq i \leq s) \\ 7 : \quad \quad R \leftarrow R - \sum_{i=1}^s Q'_i P_i \\ 8 : \quad \quad Q_i \leftarrow Q_i + Q'_i \quad (1 \leq i \leq s) \\ 9 : \quad \} \\ 10 : \quad \text{return } [Q_i, R] \end{array}$$

Here, $\text{ord}_e(R)$ is e -order of R , defined by

$$\text{ord}_e(R) = \max \{e \cdot \alpha \mid \alpha \in \text{Exps}(R)\},$$

and the total symbol of $P(x, y, \partial) = \sum_{\alpha} a_{\alpha}(x, y) \partial^{\alpha}$ ($a_{\alpha}(x, y) \in K[[x]][y]$) is $P(x, y, \xi) = \sum_{\alpha} a_{\alpha}(x, y) \xi^{\alpha}$. For example, when $P = (x_1 + x_1 x_2) \partial_1^2 + x_1 \partial_1 + 1$, $\text{ord}_e(P) = 2$ and the total symbol $P(x, \xi) = (x_1 + x_1 x_2) \xi_1^2 + x_1 \xi_1 + 1$.

Since the procedure uses $K[[x]]$ -division at the 5th line, it does not stop in finite steps. We suppose that inputs are algebraic data, in other words, inputs are elements in D . By replacing the $K[[x]]$ -division with $K[[x]]$ -approximate-division (Algorithm 1), we can get approximations of the quotients and the remainder by $\widehat{\mathcal{D}}$ -division.

Algorithm 2 ($\widehat{\mathcal{D}}$ -approximate-division).

$\widehat{\mathcal{D}}$ -approximate-division(P, P_1, \dots, P_s, N)

Input : $P, P_1, \dots, P_s \in D[y]$, $N \in \mathbb{Z}_{>0}$
Output : $\overline{Q_1}, \dots, \overline{Q_s}, \overline{R} \in D[y]$ satisfies
 $P = \overline{Q_1}P_1 + \dots + \overline{Q_s}P_s + \overline{R}$
Terms of $\overline{Q_i}$ whose total degree is smaller than $N - |\text{LE}_{<}(P_i)|$ agree with those of Q_i
Terms of \overline{R} whose total degree is smaller than N agree with those of R
Here, Q_i and R are the result of $\widehat{\mathcal{D}}$ -division(P, P_1, \dots, P_s).

```

1 :    $\overline{Q_i} \leftarrow 0$  ( $1 \leq i \leq s$ ),  $\overline{R} \leftarrow P$ 
2 :    $m_0 \leftarrow \text{ord}_e(\overline{R})$ 
3 :   for ( $k \leftarrow 0$ ;  $k \leq m_0$ ;  $k \leftarrow k + 1$ )
4 :      $M_k \leftarrow \max(|\text{LE}_{<}(P_i)| + 2(k - \text{ord}_e(P_i)) \mid i \text{ such that } k \geq \text{ord}_e(P_i))$ 
5 :     Bound  $\leftarrow N + \sum_{i=0}^{m_0} M_i$ 
6 :     for ( $k \leftarrow m_0$ ;  $k \geq 0$ ;  $k \leftarrow k - 1$ ) {
7 :        $\overline{r} \leftarrow$  (total symbol of the part of  $\overline{R}$  whose  $e$ -order is  $k$  and whose
          total degree is smaller than Bound)
8 :        $[q'_i, \overline{r}'] \leftarrow K[[x]]$ -approximate-division( $\overline{r}, \text{in}_e(P_1), \dots, \text{in}_e(P_s), <_{\overline{r}},$ 
          Bound)
9 :        $\overline{Q'_i} \leftarrow$  (replace  $\xi$  with  $\partial$  in  $q'_i$ ) ( $1 \leq i \leq s$ )
10 :       $\overline{R} \leftarrow \overline{R} - \sum_{i=0}^s \overline{Q'_i}P_i$ 
11 :       $\overline{Q_i} \leftarrow \overline{Q_i} + \overline{Q'_i}$  ( $1 \leq i \leq s$ )
12 :      Bound  $\leftarrow$  Bound  $- M_k$ 
13 :    }
14 :   return  $[\overline{Q_i}, \overline{R}]$ 
```

Since $K[[x]]$ -approximate-division stops in finite steps, the algorithm stops in finite steps. In the remainder of this section, we will prove the correctness of the algorithm.

We put

$$D(m) = \{P \in D[y] \text{ (or } \widehat{\mathcal{D}}[y]) \mid \text{ord}_e(P) = m\},$$

$$T(i) = \{P \in D[y] \text{ (or } \widehat{\mathcal{D}}[y], K[x, \xi], K[[x]][y, \xi]) \mid (\text{the total degree of every term of } P) \geq i\}.$$

It follows from the definition that $P - Q \in T(i)$ is equivalent to that P and Q are the same up to total degree $i - 1$. From the Leibnitz rule, we get the following property of the multiplication of approximate elements.

Lemma 5 (Multiplication of Approximate Elements).

$$(T(i) \cap D(m)) \cdot T(j) \subset T(i + j - 2m).$$

Here, dot \cdot means multiplication in the ring of differential operators.

Proof. We suppose that $P \in T(i) \cap D(m)$ and $Q \in T(j)$. Then we have

$$P = \sum_{|\beta| \leq m, |\alpha| + |\beta| \geq i} a_{\alpha\beta} x^\alpha \partial^\beta, \quad Q = \sum_{|\gamma| + |\delta| \geq j} b_{\gamma\delta} x^\gamma \partial^\delta \quad (a_{\alpha\beta}, b_{\gamma\delta} \in K).$$

To compute the multiplication PQ , we use the Leibnitz rule. We can get the following total symbol for PQ .

$$\begin{aligned} (PQ)(x, \xi) &= \sum_v \frac{1}{v!} \frac{\partial^{|\nu|}}{\partial \xi^\nu} \left(\sum_{(\alpha, \beta) \in \text{Exps}(P)} a_{\alpha\beta} x^\alpha \xi^\beta \right) \frac{\partial^{|\nu|}}{\partial x^\nu} \left(\sum_{(\gamma, \delta) \in \text{Exps}(Q)} b_{\gamma\delta} x^\gamma \xi^\delta \right) \\ &= \sum_{(\alpha, \beta) \in \text{Exps}(P), (\gamma, \delta) \in \text{Exps}(Q), \nu \leq \beta, \nu \leq \gamma} c_{\alpha\beta\gamma\delta\nu} x^{\alpha+\gamma-\nu} \xi^{\beta+\delta-\nu} \quad (c_{\alpha\beta\gamma\delta\nu} \in K). \end{aligned}$$

Here, $\nu \leq \beta$ means $\nu_i \leq \beta_i$ for all i .

We consider the minimum total degree of terms of PQ .

$$\min(|\alpha| + |\gamma| - |\nu| + |\beta| + |\delta| - |\nu| \mid (\alpha, \beta) \in \text{Exps}(P), (\gamma, \delta) \in \text{Exps}(Q), \nu \leq \beta, \nu \leq \gamma). \quad (1)$$

Since $|\alpha| + |\beta| \geq i$, $|\gamma| + |\delta| \geq j$ and $|\nu| \leq |\beta| \leq \text{ord}_e(P) = m$ hold, the minimum total degree (1) is more than or equal to $i + j - 2m$. Therefore, we conclude $PQ \in T(i + j - 2m)$. \square

Lemma 6. Let q_i and r be the quotients and remainder of $K[[x]]$ -division($f, g_1, \dots, g_s, <_r$). If $f \in T(N)$, then $q_i \in T(N - |\text{LE}_{<_r}(g_i)|)$, $r \in T(N)$.

Proof. It holds that

$$\begin{aligned} f &= q_1 g_1 + \dots + q_s g_s + r \\ \text{LE}_{<_r}(g_i) + \text{Exps}(q_i) &\subset \Delta^i \\ \text{Exps}(r) &\subset \bar{\Delta} \end{aligned}$$

where $\Delta^1, \dots, \Delta^s, \bar{\Delta}$ are the partition of $(\mathbb{Z}_{\geq 0})^{2n+1}$ with respect to $(\text{LE}_{<_r}(g_1), \dots, \text{LE}_{<_r}(g_s))$. From these properties, we get

$$\begin{aligned} \text{LE}_{<_r}(q_i g_i) &\leq_r \text{LE}_{<_r}(f) \\ \text{LE}_{<_r}(r) &\leq_r \text{LE}_{<_r}(f). \end{aligned}$$

From the definition of the monomial order $<_r$, we have

$$|\text{LE}_{<_r}(q_i)| \geq N - |\text{LE}_{<_r}(g_i)| \Leftrightarrow q_i \in T(N - |\text{LE}_{<_r}(g_i)|)$$

and

$$|\text{LE}_{<_r}(r)| \geq |\text{LE}_{<_r}(f)| = N \Leftrightarrow r \in T(N). \quad \square$$

Lemma 7. Let $f, g_1, \dots, g_s \in K[[x]]$ and \bar{f} be the part of f whose total degree is less than N , that is, $f - \bar{f} \in T(N)$. Let q_i and r be the quotients and remainder of $K[[x]]$ -division($f, g_1, \dots, g_s, <_r$). And let \bar{q}_i and \bar{r} be the quotients and remainder of $K[[x]]$ -approximate-division($\bar{f}, g_1, \dots, g_s, <_r, N$). Then $q_i - \bar{q}_i \in T(N - |\text{LE}_{<_r}(g_i)|)$, $r - \bar{r} \in T(N)$.

Proof. Let q'_i and r' be the quotients and remainder of $K[[x]]$ -division($\bar{f}, g_1, \dots, g_s, <_r$). Since $K[[x]]$ -division uniquely gives the quotient and remainder, the quotients and remainder of $K[[x]]$ -division($f - \bar{f}, g_1, \dots, g_s, <_r$) are $q_i - q'_i$ and $r - r'$. From $f - \bar{f} \in T(N)$ and Lemma 6, $q_i - q'_i \in T(N - |\text{LE}_{<_r}(g_i)|)$ and $r - r' \in T(N)$ holds.

From the property of $K[[x]]$ -approximate-division, $q'_i - \bar{q}_i \in T(N - |\text{LE}_{<_r}(g_i)|)$ and $r' - \bar{r} \in T(N)$ hold. So we get $q_i - \bar{q}_i \in T(N - |\text{LE}_{<_r}(g_i)|)$ and $r - \bar{r} \in T(N)$. \square

Proof of Algorithm 2. We will compare steps in the for-loop in $\widehat{\mathcal{D}}$ -division with those in $\widehat{\mathcal{D}}$ -approximate division. We put

$$M_k = \max(|\text{LE}_{<}(P_i)| + 2(k - \text{ord}_e(P_i)) \mid i \text{ such that } k \geq \text{ord}_e(P_i)).$$

We suppose that $R - \bar{R} \in T(N)$ in the $(k - 1)$ -th step. Let us compare k -th steps in the for-loop in $\widehat{\mathcal{D}}$ -division and $\widehat{\mathcal{D}}$ -approximate-division.

| |
|--|
| $\widehat{\mathcal{D}}$ -division |
| $r \leftarrow$ (total symbol of the part of R whose e -order is k) |
| $[q'_i, r'] \leftarrow K[[x]]\text{-division}(r, \text{in}_e(P_1), \dots, \text{in}_e(P_s), <_r)$ |
| $Q'_i \leftarrow q'_i(x, \partial)$ |
| $R_{\text{new}} \leftarrow R - \sum Q'_i P_i$ |
| $\widehat{\mathcal{D}}$ -approximate-division |
| $\bar{r} \leftarrow$ (total symbol of the part of \bar{R} whose e -order is k and total degree is less than N) |
| $[\bar{q}'_i, \bar{r}'] \leftarrow K[[x]]\text{-approximate-division}(\bar{r}, \text{in}_e(P_1), \dots, \text{in}_e(P_s), <_r, N)$ |
| $\bar{Q}'_i \leftarrow \bar{q}'_i(x, \partial)$ |
| $\bar{R}_{\text{new}} \leftarrow \bar{R} - \sum \bar{Q}'_i P_i$ |

We will prove that $R_{\text{new}} - \bar{R}_{\text{new}} \in T(N - M_k)$ holds after these computations have been done.

It follows from Lemma 7 that

$$\begin{aligned} q'_i - \bar{q}'_i &\in T(N - |\text{LE}_{<_r}(\text{in}_e(P_i))|) = T(N - |\text{LE}_{<}(P_i)|), \\ r' - \bar{r}' &\in T(N). \end{aligned}$$

Remainders r and \bar{r} are e -homogeneous elements of order k , and $\text{in}_e(P_i)$ is e -homogeneous element of order $\text{ord}_e(P_i)$. So, if $q'_i, \bar{q}'_i \neq 0$, then q'_i, \bar{q}'_i are e -homogeneous elements of order $k - \text{ord}_e(P_i) (\geq 0)$. And, if $r', \bar{r}' \neq 0$, then r', \bar{r}' are e -homogeneous elements of order k . Therefore, if $Q'_i, \bar{Q}'_i \neq 0$, then Q'_i, \bar{Q}'_i have the same e -order $k - \text{ord}_e(P_i)$ and are the same up to total degree $N - |\text{LE}_{<}(P_i)| - 1$. In other words, it holds that

$$Q'_i - \bar{Q}'_i \in T(N - |\text{LE}_{<}(P_i)|) \cap D(k - \text{ord}_e(P_i)).$$

From Lemma 5, we have

$$(T(N - |\text{LE}_{<}(P_i)|) \cap D(k - \text{ord}_e(P_i))) \cdot T(0) \subset T(N - |\text{LE}_{<}(P_i)| - 2(k - \text{ord}_e(P_i))).$$

Therefore, it holds that

$$Q'_i P_i - \bar{Q}'_i P_i \in T(N - |\text{LE}_{<}(P_i)| - 2(k - \text{ord}_e(P_i))).$$

Since $M_k = \max(|\text{LE}_{<}(P_i)| + 2(k - \text{ord}_e(P_i)) \mid i \text{ such that } k \geq \text{ord}_e(P_i))$, we get

$$\sum_{i=1}^s Q'_i P_i - \sum_{i=1}^s \bar{Q}'_i P_i \in T(N - M_k).$$

Therefore, $R_{\text{new}} - \bar{R}_{\text{new}} \in T(N - M_k)$.

The accuracy of approximation decreases by M_k at each step of the for loop. To keep the accuracy, we beforehand add $\sum_{i=0}^{m_0} M_i$ to N . \square

Example 8 (Example of $\widehat{\mathcal{D}}$ -Division and $\widehat{\mathcal{D}}$ -Approximate-Division). Let $n = 1$. We put $P = \partial^2$, $P_1 = (1 + x)\partial + x$. We compare $\widehat{\mathcal{D}}$ -division(P, P_1) and $\widehat{\mathcal{D}}$ -approximate-division($P, P_1, 5$).

At first, we show the procedure of $\widehat{\mathcal{D}}$ -division(P, P_1).

$$\begin{array}{l}
\hline
R \leftarrow P, m_0 \leftarrow \text{ord}_e(R) = 2 \\
r \leftarrow (\text{part of } R \text{ whose } e\text{-order is } 2) = \xi^2 \\
[q'_1, r'] \leftarrow K[[x]]\text{-division}(r, \text{in}_e(P_1), <_r) \\
(q'_1 = \frac{1}{1+x}\xi, r' = 0) \\
Q'_1 \leftarrow \frac{1}{1+x}\partial \\
R \leftarrow R - Q'_1P_1 = -\partial - \frac{1}{1+x} \\
\hline
r \leftarrow (\text{part of } R \text{ whose } e\text{-order is } 1) = -\xi \\
[q'_1, r'] \leftarrow K[[x]]\text{-division}(r, \text{in}_e(P_1), <_r) \\
(q'_1 = -\frac{1}{1+x}, r' = 0) \\
Q'_1 \leftarrow -\frac{1}{1+x} \\
R \leftarrow R - Q'_1P_1 = \frac{-1+x}{1+x} \\
\hline
r \leftarrow (\text{part of } R \text{ whose } e\text{-order is } 0) = \frac{-1+x}{1+x} \\
[q'_1, r'] \leftarrow K[[x]]\text{-division}(r, \text{in}_e(P_1), <_r) \\
(q'_1 = 0, r' = \frac{-1+x}{1+x}) \\
Q'_1 \leftarrow 0 \\
R \leftarrow R - Q'_1P_1 = \frac{-1+x}{1+x} \\
\hline
\end{array}$$

The output of $\widehat{\mathcal{D}}\text{-division}(P, P_1)$ is

$$\partial^2 = \left(\frac{1}{1+x}\partial - \frac{1}{1+x} \right) ((1+x)\partial + x) + \frac{-1+x}{1+x}.$$

The quotient is

$$\frac{1}{1+x}\partial - \frac{1}{1+x} = (1-x+x^2-x^3+x^4-\dots)(\partial-1)$$

and the remainder is

$$\frac{-1+x}{1+x} = -1 + 2x - 2x^2 + 2x^3 - 2x^4 + 2x^5 - \dots$$

Next, we show the procedure of $\widehat{\mathcal{D}}\text{-approximate-division}(P, P_1, 5)$.

$$\begin{array}{l}
\hline
\bar{R} \leftarrow P, m_0 \leftarrow \text{ord}_e(R) = 2 \\
M_0 \leftarrow 0, M_1 \leftarrow 1, M_2 \leftarrow 3 \\
\text{Bound} \leftarrow 5 + (0 + 1 + 3) = 9 \\
\hline
\bar{r} \leftarrow (\text{part of } \bar{R} \text{ whose } e\text{-order is } 2 \text{ and total degree is less than } 9) = \xi^2 \\
[\bar{q}'_1, \bar{r}'] \leftarrow K[[x]]\text{-approximate-division}(\bar{r}, \text{in}_e(P_1), <_r, 9) \\
(\bar{q}'_1 = (1-x+x^2-\dots+x^6)\xi, \bar{r}' = -x^7\xi^2) \\
\bar{Q}'_1 \leftarrow (1-x+x^2-\dots+x^6)\partial \\
\bar{R} \leftarrow \bar{R} - \bar{Q}'_1P_1 = -x^7\partial^2 - \partial - x^7\partial - 1 + x - x^2 + x^3 - x^4 + x^5 - x^6 \\
\text{Bound} \leftarrow 9 - 3 = 6 \\
\hline
\bar{r} \leftarrow (\text{part of } \bar{R} \text{ whose } e\text{-order is } 1 \text{ and total degree is less than } 6) = -\xi \\
[\bar{q}'_1, \bar{r}'] \leftarrow K[[x]]\text{-approximate-division}(\bar{r}, \text{in}_e(P_1), <_r, 6) \\
(\bar{q}'_1 = -1 + x - x^2 + x^3 - x^4, \bar{r}' = x^5\xi) \\
\bar{Q}'_1 \leftarrow -1 + x - x^2 + x^3 - x^4 \\
\bar{R} \leftarrow \bar{R} - \bar{Q}'_1P_1 = -x^7\partial^2 + x^5\partial - x^7\partial - 1 + 2x - 2x^2 + 2x^3 - 2x^4 + 2x^5 - x^6 \\
\text{Bound} \leftarrow 6 - 1 = 5 \\
\hline
\bar{r} \leftarrow (\text{part of } \bar{R} \text{ whose } e\text{-order is } 0 \text{ and total degree is less than } 5) \\
= -1 + 2x - 2x^2 + 2x^3 - 2x^4 \\
[\bar{q}'_1, \bar{r}'] \leftarrow K[[x]]\text{-approximate-division}(\bar{r}, \text{in}_e(P_1), <_r, 5) \\
(\bar{q}'_1 = 0, \bar{r}' = \bar{r} = -1 + 2x - 2x^2 + 2x^3 - 2x^4) \\
\bar{Q}'_1 \leftarrow 0 \\
\bar{R} \leftarrow \bar{R} = -x^7\partial^2 + x^5\partial - x^7\partial - 1 + 2x - 2x^2 + 2x^3 - 2x^4 + 2x^5 - x^6 \\
\hline
\end{array}$$

The output of $\widehat{\mathcal{D}}$ -approximate-division($P, P_1, 5$) is

$$\partial^2 = ((1 - x + x^2 + \cdots + x^6)\partial + (-1 + x - x^2 + x^3 - x^4)) \cdot ((1 + x)\partial + x) + \\ - x^7\partial^2 - x^7\partial + x^5\partial - 1 + 2x - 2x^2 + 2x^3 - 2x^4 + 2x^5 - x^6.$$

The correct part of the remainder is

$$-1 + 2x - 2x^2 + 2x^3 - 2x^4.$$

And this part is the same with the part of the remainder of $\widehat{\mathcal{D}}$ -division.

4. Computation of the local b function by the approximate division algorithm in $\widehat{\mathcal{D}}[s]$

We will apply the approximate division algorithm in $\widehat{\mathcal{D}}$ (Algorithm 2) to obtain the local b function of a polynomial. In the sequel, we use the parameter variable s instead of y , and K denotes the field of complex numbers.

4.1. Definition and algorithm of the b function

Definition 9 (Global b Function). For given $f \in K[x]$, we define the global b function as the monic polynomial $\tilde{b}(s)$ of the least degree which satisfies $\exists P \in D[s]$ s.t. $P \cdot f^{s+1} = \tilde{b}(s)f^s$.

Definition 10 (Local b Function). For given $f \in K[x]$, we define the local b function at the origin as the monic polynomial $b(s)$ of the least degree which satisfies $\exists P \in \widehat{\mathcal{D}}[s]$ s.t. $P \cdot f^{s+1} = b(s)f^s$.

Example 11 (b Function). Let $f = x(x + y + 1) = x^2 + xy + x$. The global b function of f is $(s + 1)^2$, and $(\partial_x^2 + \partial_x \partial_y) \cdot f^{s+1} = (s + 1)^2 f^s$ holds. The local b function of f is $s + 1$, and $\frac{1}{1+2x+y} \partial_x \cdot f^{s+1} = (s + 1)f^s$ holds.

Oaku gave algorithms computing the global b function and the local b function for a given polynomial (Oaku, 1997b,a; Oaku and Takayama, 2001). A Gröbner basis method in D is used in these algorithm.

Noro gave an efficient algorithm computing the global b function (Noro, 2002). In this algorithm, a Gröbner basis method in D , and modular method are used. Especially, to eliminate variables, normal forms with respect to a Gröbner basis are used.

In this paper, we propose an algorithm computing the local b function by utilizing an approximate normal form with respect to a Gröbner basis. Our algorithm is an analogous algorithm with Noro's algorithm. For this purpose, we review a Gröbner basis and a normal form in $\widehat{\mathcal{D}}$ and define an approximate normal form.

4.2. Gröbner basis and normal form in $\widehat{\mathcal{D}}[s]$

Although $\widehat{\mathcal{D}}[s]$ is a transcendental object, for ideals in $\widehat{\mathcal{D}}[s]$ generated by elements in $D[s]$, we can compute a Gröbner basis in finite steps by either of the following algorithms.

- Lazard's method in D (using a homogenization)
- method using the Mora division algorithm in D (Granger and Oaku, 2004; Granger et al., 2005)

Definition 12 (Normal Form in $\widehat{\mathcal{D}}[s]$). For $P \in \widehat{\mathcal{D}}[s]$ and a Gröbner basis G in $\widehat{\mathcal{D}}[s]$, the remainder of $\widehat{\mathcal{D}}$ -division(P, G) (Theorem 4) is uniquely determined. We call the remainder the normal form of P by G with respect to the monomial order $<$, and we denote it by $\text{NF}(P, G, <)$.

We note that a normal form does not have a finite representation in general.

Lemma 13 (Ideal Membership). Let G be a Gröbner basis of an ideal \mathcal{I} in $\widehat{\mathcal{D}}[s]$. For $P \in \widehat{\mathcal{D}}[s]$, $P \in \mathcal{I}$ is equivalent to $\text{NF}(P, G, <) = 0$.

Lemma 14 (Sum of Normal Forms). Let G be a Gröbner basis in $\widehat{\mathcal{D}}[s]$. For $P, Q \in \widehat{\mathcal{D}}[s]$, it holds that

$$\text{NF}(P + Q, G, <) = \text{NF}(P, G, <) + \text{NF}(Q, G, <).$$

4.3. Algorithm computing $\mathcal{I} \cap K[s]$

For an ideal \mathcal{I} in $\widehat{\mathcal{D}}[s]$, we propose an algorithm computing a generator of $\mathcal{I} \cap K[s]$. In this algorithm, we use approximate normal forms. We fix a Gröbner basis G of \mathcal{I} .

For $g(s) = a_l s^l + a_{l-1} s^{l-1} + \cdots + a_0 \in \mathcal{I}$, it holds that

$$g(s) \in \mathcal{I} \Leftrightarrow \text{NF}(g(s), G, <) = 0 \\ \Leftrightarrow a_l \text{NF}(s^l, G, <) + a_{l-1} \text{NF}(s^{l-1}, G, <) + \cdots + a_0 \text{NF}(1, G, <) = 0.$$

We compute $\text{NF}(s^i, G, <)$ beforehand, and we compute the minimum l and coefficients $a_l, \dots, a_0 \in K$, $a_l \neq 0$ which satisfies

$$a_l \text{NF}(s^l, G, <) + a_{l-1} \text{NF}(s^{l-1}, G, <) + \cdots + a_0 \text{NF}(1, G, <) = 0.$$

Then the polynomial $a_l s^l + \cdots + a_0$ is a generator of the intersection.

However, we cannot generally compute normal forms in $\widehat{\mathcal{D}}[s]$ in finite steps. The normal form $\text{NF}(s^l, G, <)$ has infinite terms in general. Therefore, we use an approximate normal form.

Definition 15 (*Approximate Normal Form*). Let $P \in \widehat{\mathcal{D}}[s]$ and G be a Gröbner basis of an ideal \mathcal{I} in $\widehat{\mathcal{D}}[s]$. We define the approximate normal form of P by G up to total degree N as the remainder of $\widehat{\mathcal{D}}$ -approximate-division(P, G, N) (Algorithm 2). We denote it by $\text{NF}(P, G, <, N)$. By definition, $\text{NF}(P, G, <)$ and $\text{NF}(P, G, <, N)$ are the same up to the total degree $N - 1$.

We suppose that

$$a_l \text{NF}(s^l, G, <, N) + a_{l-1} \text{NF}(s^{l-1}, G, <, N) + \cdots + a_0 \text{NF}(1, G, <, N) = 0.$$

Since we use approximate normal forms, it is not always true that $a_l s^l + a_{l-1} s^{l-1} + \cdots + a_0 \in \mathcal{I}$. However, we can solve an ideal membership problem for algebraic data in $\widehat{\mathcal{D}}[s]$ by utilizing the Mora division algorithm in $D[s]$. Therefore, we need to apply the Mora division algorithm to the gotten candidate $a_l s^l + \cdots + a_0$ in order to check if it is a member of \mathcal{I} .

We will explain an algorithm computing $\mathcal{I} \cap K[s]$ by using these ideas. We suppose $\mathcal{I} \cap K[s] \neq \{0\}$. We put

$$L_{i,N} = \{(a_0, \dots, a_i) \in K^{i+1} \mid a_i \text{NF}(s^i, G, <, N) + \cdots + a_0 \text{NF}(1, G, <, N) = 0\} \\ L_i = \{(a_0, \dots, a_i) \in K^{i+1} \mid a_i \text{NF}(s^i, G, <) + \cdots + a_0 \text{NF}(1, G, <) = 0\},$$

where G is a Gröbner basis of \mathcal{I} and $N \in \mathbb{N}$. It holds that

$$L_{i,0} \supset L_{i,1} \supset L_{i,2} \supset \cdots \supset L_i.$$

The polynomial $a_i s^i + \cdots + a_0$ which satisfies $L_i \neq \{0\}$, $L_{i-1} = \{0\}$ and $(a_0, \dots, a_i) \in L_i$ is a generator of $\mathcal{I} \cap K[s]$. In summary, an algorithm to compute the generator is as follows:

Algorithm 3 (*Computing the Generator of $\mathcal{I} \cap K[s]$*).

- (1) $N \leftarrow$ (a natural number), $d \leftarrow 0$.
- (2) Find i which satisfies $L_{i-1,N} = \{0\}$, $L_{i,N} \neq \{0\}$. (Compute $L_{d,N}, L_{d+1,N}, \dots$. Note that $L_{i-1,N} = \{0\} \Rightarrow L_0 = L_1 = \cdots = L_{i-1} = \{0\}$.)
- (3) Take an element $(a_i, \dots, a_0) \in L_{i,N}$ and put $g(s) = a_i s^i + \cdots + a_0$.
- (4) Divide $g(s)$ by G by using the Mora division algorithm. If the remainder is 0, then $g(s)$ is the generator. If not, then set $N \leftarrow N + 1$, $d \leftarrow i$ and goto (2).

4.4. Algorithm computing the local b function

For a polynomial $f \in K[x]$, an algorithm of computing the local b function of f at the origin is as follows:

- (1) Compute a set of generators of $\text{Ann}_{\widehat{\mathcal{D}}[s]} f^s$. Denote it by H .
- (2) Set $\mathcal{I} = \widehat{\mathcal{D}}[s](H \cup \{f\})$. Compute the generator $b(s)$ of $\mathcal{I} \cap K[s]$. The polynomial $b(s)$ is the local b function.

In (1), we take a set of generators of $\text{Ann}_{D[s]} f^s$ as a set of generators of $\text{Ann}_{\widehat{\mathcal{D}}[s]} f^s$. We can compute a set of generators of $\text{Ann}_{D[s]} f^s$ by Oaku's algorithm (Oaku, 1997a). In (2), since $\mathcal{I} \cap K[s] \neq \{0\}$ holds (from the existence of the local b function), we can utilize the Algorithm 3.

We show an example of computing the local b function.

Example 16 (Local b Function of $f = x^2(y+1)^2z^2$ at the Origin). The global b function of f is $(s+1)^3(s+\frac{1}{2})^3$, and the local b function is $(s+1)^2(s+\frac{1}{2})^2$.

A set of generators H of $\text{Ann}_{\widehat{\mathcal{D}}[s]} f^s$ is

$$\{P_1 = -2s + z\partial_z, P_2 = -x\partial_x + z\partial_z, P_3 = -\partial_y + x\partial_x - y\partial_y\}.$$

A Gröbner basis G of $\mathcal{J} = \widehat{\mathcal{D}}[s](H \cup \{f\})$ with respect to $<$ is

$$\{f, P_1, P_2, P_3, P_4 = -xz^3\partial_z - 2xz^2, P_5 = -z^4\partial_z^2 - 2xz^2\partial_x - 2z^3\partial_z - 2z^2\}.$$

We set $N = 1$ and will apply the Algorithm 3 to compute the intersection $\mathcal{J} \cap K[s]$. We note that approximate normal forms of s^i by G up to total degree $N = 7$ are

$$\text{NF}(1, G, <, 7) = 1$$

$$\text{NF}(s, G, <, 7) = \frac{1}{2}z\partial_z$$

$$\text{NF}(s^2, G, <, 7) = \frac{1}{4}z^2\partial_z^2 + \frac{1}{4}z\partial_z$$

$$\text{NF}(s^3, G, <, 7) = \frac{1}{8}z^3\partial_z^3 + \frac{3}{8}z^2\partial_z^2 + \frac{1}{8}z\partial_z$$

$$\text{NF}(s^4, G, <, 7) = -\frac{3}{8}z^3\partial_z^3 - \frac{31}{16}z^2\partial_z^2 - \frac{31}{16}z\partial_z - \frac{1}{4}.$$

Steps of Algorithm 3 are

| | |
|--|---|
| $L_{1,1} = \{0\}$ | $L_1 = \{0\}$ |
| $L_{2,1} \ni (0, 1)$ | $s \xrightarrow[\text{G}]{\text{Mora division}} \text{non-zero}$ |
| $L_{2,2} \ni (0, 1)$ | $s \xrightarrow[\text{G}]{\text{Mora division}} \text{non-zero}$ |
| $L_{2,3} = \{0\}$ | $L_2 = \{0\}$ |
| $L_{3,3} \ni (0, -\frac{1}{2}, 1)$ | $s^2 - \frac{1}{2}s \xrightarrow[\text{G}]{\text{Mora division}} \text{non-zero}$ |
| $L_{3,4} \ni (0, -\frac{1}{2}, 1)$ | $s^2 - \frac{1}{2}s \xrightarrow[\text{G}]{\text{Mora division}} \text{non-zero}$ |
| $L_{3,5} = \{0\}$ | $L_3 = \{0\}$ |
| $L_{4,5} \ni (0, \frac{1}{2}, -\frac{3}{2}, 1)$ | $s^3 - \frac{3}{2}s^2 + \frac{1}{2}s \xrightarrow[\text{G}]{\text{Mora division}} \text{non-zero}$ |
| $L_{4,6} \ni (0, \frac{1}{2}, -\frac{3}{2}, 1)$ | $s^3 - \frac{3}{2}s^2 + \frac{1}{2}s \xrightarrow[\text{G}]{\text{Mora division}} \text{non-zero}$ |
| $L_{4,7} = \{0\}$ | $L_4 = \{0\}$ |
| $L_{5,7} \ni (\frac{1}{4}, \frac{3}{2}, \frac{13}{4}, 3, 1)$ | $s^4 + 3s^3 + \frac{13}{4}s^2 + \frac{3}{2}s + \frac{1}{4} \xrightarrow[\text{G}]{\text{Mora division}} 0.$ |

Therefore, the local b function is $s^4 + 3s^3 + \frac{13}{4}s^2 + \frac{3}{2}s + \frac{1}{4} = (s+1)^2(s+\frac{1}{2})$.

5. Implementation and timing data

Our algorithm computing the local b function has been implemented by utilizing computer algebra system “Risa/Asir”. The name of our package is “nk_mora/local-b.rr”. We show the timing data.

Our algorithm computing the local b function consists of the following 3 parts.

- (1) Computation of a set of generators of $\text{Ann}_{\widehat{\mathcal{D}}[s]} f^s$. (In fact, computation of a set of generators of $\text{Ann}_{D[s]} f^s$.)
- (2) Computation of the Gröbner basis of $\mathcal{I} = \text{Ann}_{\widehat{\mathcal{D}}[s]} f^s + \widehat{\mathcal{D}}[s]f$.
- (3) Computation of the intersection $\mathcal{I} \cap K[s]$ (Algorithm 3).

To perform the step (1), we use Oaku’s algorithm computing $\text{Ann}_{D[s]} f^s$ (Oaku, 1997b). In the timing table, we denote this by (\mathcal{I}) . To perform the step (2), we have the following 2 algorithms.

- (2a) Buchberger algorithm utilizing the Mora division algorithm in $D[s]$.
- (2b) Lazard’s homogenized method in $D[s]$.

We denote this parts by **(GB-a)** and **(GB-b)**. To perform the step (3), we use the Algorithm 3. We denote this part by **(localb-nf)**.

We took some examples from Oaku (1997a).

| f | \mathcal{I} | GB-a | GB-b | localb-nf | localb | deg |
|--------------------------|---------------|-----------------|-----------------|-----------------|---------|-----|
| $x + y^2 + z^2$ | 0.00504 | 0.0190 | 0.00494 | 0.00503 | 0.00612 | 1 |
| $x^2 + y^2 + z^2$ | 0.00524 | 0.00157 | 0.00224 | 0.0229 | 0.0119 | 2 |
| $x^3 + y^2 + z^2$ | 0.00637 | 0.0298 | 0.0694 + 0.0192 | 0.0417 + 0.0244 | 0.0226 | 3 |
| $x^4 + y^2 + z^2$ | 0.00631 | 0.0317 | 1.41 + 0.27 | 0.131 + 0.0884 | 0.0384 | 4 |
| $x^5 + y^2 + z^2$ | 0.00624 | 0.0306 | 7.23 + 1.51 | 0.325 + 0.272 | 0.0612 | 5 |
| $x^6 + y^2 + z^2$ | 0.00619 | 0.0291 | 33.0 + 3.25 | 0.726 + 0.598 | 0.129 | 6 |
| $x^7 + y^2 + z^2$ | 0.00610 | 0.0281 + 0.0170 | 137.5 + 24.1 | 1.26 + 0.674 | 0.247 | 7 |
| $x^3 + xy^2 + z^2$ | 0.0260 | 0.172 | 0.107 | 0.114 + 0.0756 | 0.0767 | 4 |
| $x^4 + xy^2 + z^2$ | 0.0566 | 0.140 | 2.29 + 0.338 | 0.501 + 0.284 | 0.184 | 6 |
| $x^5 + xy^2 + z^2$ | 0.0903 | 0.194 | 32.6 + 5.47 | 0.505 + 0.284 | 0.173 | 6 |
| $x^3 + y^4 + z^2$ | 0.00858 | 0.118 | 1.45 | 1.43 + 0.821 | 0.197 | 7 |
| $x^3 + xy^3 + z^2$ | 0.0110 | 0.236 + 0.337 | 1.09 | 2.94 + 1.40 | 1.06 | 8 |
| $x^3 + y^5 + z^2$ | 0.00979 | 0.123 | 18.3 + 2.82 | 4.36 + 1.80 | 0.615 | 9 |
| $x^5 + x^3y^3 + y^5$ | 0.70 | 0.28 | 0.052 | — | 7.06 | 7 |
| $x^4 + x^2y^2 + y^4$ | 0.0061 | 0.0138 | 0.0019 | 0.163 | 0.153 | 6 |
| $x^3 + x^2y^2 + y^3$ | 0.031 | 0.0748 | 0.018 | 33.4 | 0.0932 | 4 |
| $x^3 + y^3 + z^3$ | 0.0062 | 0.00162 | 0.0025 | 0.142 | 0.112 | 5 |
| $x^6 + y^4 + z^3$ | 0.0134 | 0.052 | 76.5 | 131.1 | 23.2 | 18 |
| $x^3 + y^2z^2$ | 0.029 | 0.0462 | 0.016 | 0.868 + 0.108 | 0.144 | 7 |
| $(x^3 - y^2z^2)^2$ | 0.102 | 7.49 + 1.67 | 0.0932 | 17.4 + 1.40 | 3.99 | 14 |
| $x^5 - y^2z^2$ | 0.013 | 0.0827 | 0.092 | 3.77 + 0.279 | 0.817 | 13 |
| $x^5 - y^2z^3$ | 0.0088 | 0.0265 | 0.0028 | 15.9 + 1.22 | 3.53 | 17 |
| $x^3 + y^3 - 3xyz$ | 0.019 | 0.473 | 0.020 | 0.641 + 0.0540 | 0.0946 | 5 |
| $x^3 + y^3 + z^3 - 3xyz$ | 0.013 | 0.140 | 0.012 | 0.0658 | 0.0207 | 3 |
| $y(x^5 - y^2z^2)$ | 0.014 | 16.7 + 3.43 | 0.58 | 58.6 + 5.24 | 6.10 | 18 |
| $y(x^3 - y^2z^2)$ | 0.644 | 2.65 + 0.668 | 0.15 | 4.18 + 0.338 | 0.711 | 10 |
| $y((y + 1)x^3 - y^2z^2)$ | 0.284 | 18.8 + 1.34 | 3.6 + 0.14 | — | 10.2 | 10 |
| $x^6 + x^3y^3 + y^6$ | 0.004 | 0.012 | 0.002 | 1.38 | 1.73 | 10 |
| $x^8 + x^4y^4 + y^8$ | 0.003 | 0.058 | 0.0039 | 5.25 | 16.3 | 14 |
| $x^5 + y^5 + z^5$ | 0.004 | 0.018 | 0.003 | 6.86 | 8.58 | 11 |
| $x^6 + y^6 + z^6$ | 0.003 | 0.018 | 0.003 | 25.2 | 56.4 | 14 |

- localb — total time of Oaku’s algorithm computing the local b function (Oaku, 1997b)
- deg — degree of the local b function
- machine — CPU: Athlon MP 1800+ (1533 MHz) (2 CPU), Memory: 3GB, OS: FreeBSD 4.8

The total time of our algorithm is $(\mathcal{I} + \text{GB-a} + \text{localb-nf})$ and the total time of Oaku’s algorithm is **localb**. Oaku’s algorithm(localb) is faster than our algorithm for the first 27 examples in the table. The reasons seems to be as follows:

- (1) $\widehat{\mathcal{D}}$ -approximate-division algorithm (Algorithm 2) is heavy computation in our theory and implementation. The variable Bound in Algorithm 2 becomes very large and $K[[x]]$ -approximate-division becomes heavy. (It is a question if we can make the Bound smaller with a sharper estimate.)
- (2) Mora division algorithm in D is heavy computation.

Although there are these negative aspects in practice in our algorithm, our algorithm is faster than Oaku's one for the last 4 examples in the table. It is a future question to find conditions so that our algorithm is faster than Oaku's one. One simple observation is that the last 4 examples are homogeneous.

Although, our new method is not always faster than Oaku's one, however the author thinks that it is promising method for some class of problems. In fact, for the last example, Oaku's method spends a lot of time to compute a Groebner basis computation in the Weyl algebra D which is done with highly optimized built-in function in "Risa/Asir". Our implementations of the Mora division in D and $\widehat{\mathcal{D}}$ -approximate-division are written in the user language of "Risa/Asir" and further reduction strategies in these algorithm have not yet been highly optimized, which seems to be an interesting research subject.

References

- Castro, F., 1984. Théorème de division pur les opérateurs différentiels et calcul des multiplicités. Ph.D. Thesis. Université Paris VII.
- Castro, F., Granger, M., 2004. Explicit calculations in rings of differential operators. Séminaires et Congrès 8, 89–128.
- Granger, M., Oaku, T., 2004. Minimal filtered free resolutions and division algorithms for analytic D -modules. Journal of Pure and Applied Algebra 191 (1–2), 157–180.
- Granger, M., Oaku, T., Takayama, N., 2005. Tangent cone algorithm for homogenized differential operators. Journal of Symbolic Computation 39, 417–431.
- Greuel, G.-M., Pfister, G., 1996. Advances and improvements in the theory of standard bases and syzygies. Archiv der Mathematik 66, 163–176.
- Kobayashi, H., Furukawa, A., Sasaki, T., 1986. Gröbner basis of ideal of convergent power series. Sūrikaiseikikenkyūsho Kōkyūroku 581, 7–25.
- Noro, M., 2002. An efficient modular algorithm for computing the global b -function. Mathematical Software (Beijing 2002) 147–157.
- Oaku, T., 1997a. Algorithm for the b -function and D -modules associated with a polynomial. Journal of Pure and Applied Algebra 117–118, 495–518.
- Oaku, T., 1997b. An algorithm of computing b -function. Duke Mathematical Journal 87 (1), 115–132.
- Oaku, T., Takayama, N., 2001. Algorithms for D -modules – restriction, tensor product, localization, and local cohomology groups. Journal of Pure and Applied Algebra 156, 267–308.